

Appl. No. 09/942,390
Amdt. Dated March 7, 2005
Reply to Office action of January 3, 2005

BEST AVAILABLE COPY

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (Currently Amended): A method of establishing a loop of delayed commands for completion among a larger set of commands in a queue of commands, comprising:

recognizing a delay in completion of a first read command in the queue and in response designating the first read command as a first delayed read command;

determining that the command in the queue following the first delayed read command is a second read command;

setting a loop start pointer to identify the first delayed read command; setting a loop end pointer to identify the second read command;

recognizing a delay in completion of the second read command in the queue and in response designating the second read command as a second delayed read command;

determining that the command in the queue following the second delayed read command is a third read command;

advancing the loop end pointer to identify the third read command; and attempting to complete the read commands at and between the loop start pointer and the loop end pointer until all of those read commands have been completed, before attempting to complete other commands in the queue; and

advancing the loop start pointer to the next subsequent command in the command buffer after the command identified by the loop start pointer has been completed.

Claim 2 (Original): A method as defined in claim 1, further comprising:

Appl. No. 09/942,390
Amdt. Dated March 7, 2005
Reply to Office action of January 3, 2005

PTO REVERSED

completing the first delayed read command; and
advancing the loop start pointer to identify the second delayed read
command after the first delayed read command has been completed.

Claim 3 (Original): A method as defined in claim 2, further comprising:
recognizing a delay in completion of the third read command and in
response designating the third read command as a third delayed read command;
determining that the command in the queue following the third read
command is a fourth read command; and
advancing the loop end pointer to identify the fourth read command.

Claim 4 (Original): A method as defined in claim 2, further comprising:
recognizing the completion of all but one of the read commands
between the loop start pointer and the loop end pointer; and
eliminating the loop start pointer and the loop end pointer upon
recognizing the completion of all but the one read command.

Claim 5 (Previously Amended): A method as defined in claim 2 wherein the
commands in the queue are selected for completion by a queue pointer, and the
method further comprises:

adjusting the position of the queue pointer to move through the
commands at and between the loop start pointer and the loop end pointer until all of
the commands at and between the loop start pointer and the loop end pointer have
been completed.

Claim 6 (Cancelled)

Claim 7 (Previously Amended): A method as defined in claim 5, further
comprising:

advancing the queue pointer in the sequence of commands starting
with the command identified by the loop start pointer and continuing through all of
the commands at and between the loop start pointer and the loop end pointer and
ending with the command identified by the loop end pointer.

Claim 8 (Original): A method as defined in claim 7, further comprising:

Appl. No. 09/942,390
Amdt. Dated March 7, 2005
Reply to Office action of January 3, 2005

marking each command identified at and between the loop start pointer and the loop end pointer as invalid upon completion of each such command.

Claim 9 (Original): A method as defined in claim 2, further comprising:

advancing the loop start pointer to identify the same command that is identified by the loop end pointer.

Claim 10 (Original): A method as defined in claim 9, further comprising:

abolishing the loop of delayed read commands by eliminating the loop start pointer and the loop end pointer upon loop start pointer identifying the same command that is identified by the loop end pointer.

Claim 11 (Original): A method as defined in claim 1, further comprising:

advancing the loop start pointer to the next subsequent command in the queue after completing the command identified by the loop start pointer.

Claim 12 (Previously Amended): A method as defined in claim 1, further comprising:

for each subsequent read command subsequent to the aforementioned first, second and third read commands, advancing the loop end pointer to the next subsequent read command in the queue upon experiencing a delay in completing the read command previously identified by the loop end pointer.

Claim 13 (Currently Amended): An interface controller for communicating commands and data to at least one target device attached to the interface controller; the interface controller comprising:

a command queue having a command buffer in which the commands are recorded;

a data mover for transferring commands to a target device and receiving responses from the target device in response to commands; and

a queue processor connected to the command queue and the data mover, the queue processor operatively applying selected commands from the command buffer to the data mover to be transferred to the target device for completion, the queue processor including programmed logic functionality which:

Appl. No. 09/942,390
Amdt. Dated March 7, 2005
Reply to Office action of January 3, 2005

applies a first read command from the command buffer to the data mover for transfer to a target device for completion;
recognizes a response from the target device indicating a delay in completion of the first read command;
determines that the command in the command buffer following the first read command is a second read command;
sets a loop start pointer to identify the first read command;
sets a loop end pointer to identify the second read command;
applies the second read command from the command buffer to the data mover for transfer to a target device for completion;
recognizes a response from the target device indicating a delay in completion of the second read command;
determines that the command in the command buffer following the second read command is a third read command;
advances the loop end pointer to identify the third read command; and
applies the third read command from the command buffer to the data mover for transfer to a target device for completion; and
advances the loop start pointer to the next subsequent command in the command buffer after the command identified by the loop start pointer has been completed.

Claim 14 (Original): An interface controller as defined in claim 13, wherein the queue processor includes further programmed logic functionality which:

identifies the read commands identified by and between the loop start pointer and the loop end pointer as delayed read commands;
establishes a loop of the delayed read commands; and
applies the delayed read commands of the loop to the data mover for transfer to the target device for completion before applying any other read commands from the command buffer to the data mover for transfer to the target device.

Appl. No. 09/942,390
Amdt. Dated March 7, 2005
Reply to Office action of January 3, 2005

Claim 15 (Previously Amended): An interface controller as defined in claim 14, wherein the queue processor includes further programmed logic functionality which:

recognizes responses from each target device indicating completion of each read command transferred to the target device;

recognizes completion of the delayed read command identified by the loop start pointer; and thereafter

advances the loop start pointer to identify the next uncompleted delayed read command of the loop.

Claim 16 (Original): An interface controller as defined in claim 15, wherein the queue processor includes further programmed logic functionality which:

advances the loop start pointer to identify the delayed read command which is also identified by the loop end pointer upon recognizing that all of the other delayed read commands in the loop have been completed; and thereafter

abolishes the loop by eliminating the loop start pointer and the loop end pointer.

Claim 17 (Original): An interface controller as defined in claim 14, wherein the queue processor includes further programmed logic functionality which:

recognizes the completion of all but a remaining one of the delayed read commands in the loop; and thereafter

abolishes the loop by eliminating the loop start pointer and the loop end pointer.

Claim 18 (Previously Amended): An interface controller as defined in claim 17, wherein the queue processor includes further programmed logic functionality which:

recognizes a response from the target device indicating completion of the remaining one delayed read command; and thereafter

Appl. No. 09/942,390
Amdt. Dated March 7, 2005
Reply to Office action of January 3, 2005

applies another command from the command buffer to the data mover for transfer to the target device for completion, the another command being a command not included in the loop before the loop was abolished.

Claim 19 (Previously Amended): An interface controller as defined in claim 14, wherein the queue processor includes further programmed logic functionality which:

applies the delayed read commands of the loop to the data mover for transfer to the target device for completion until the loop of delayed read commands is abolished, before applying any other read commands from the command buffer to the data mover for transfer to the target device.

Claim 20 (Cancelled)

Claim 21 (Original): An interface controller as defined in claim 13, wherein the commands of the command buffer are normally selected for completion by queue positioning functionality of the interface controller which positions a queue pointer to identify the command of the command buffer selected for completion, and the queue processor includes further programmed logic functionality which:

overrides the normal positioning of the queue positioning functionality of the interface controller; and

limits the position of the queue pointer to those commands identified by and between the loop start pointer and the loop end pointer until all of the uncompleted read commands identified between the loop start pointer and the loop end pointer have been completed.

Claim 22 (Original): An interface controller as defined in claim 21, wherein the queue processor includes further programmed logic functionality which:

marks each delayed read command identified by and between the loop start pointer and the loop end pointer as invalid upon completion of that command.

Claim 23 (Original): An interface controller as defined in claim 13, wherein the queue processor includes further programmed logic functionality which:

advances the loop start pointer to the next subsequent command in the

Appl. No. 09/942,390
Amdt. Dated March 7, 2005
Reply to Office action of January 3, 2005

Office action of January 3, 2005

command buffer after the command identified by the loop start pointer has been completed.

Claim 24 (Cancelled)

Claim 25 (Original): An interface controller as defined in claim 13, wherein the queue processor includes further programmed logic functionality which:

 determines that the command in the command buffer following the third read command is a fourth read command;

 advances the loop end pointer to identify the fourth read command;
and

 applies the fourth read command from the command buffer to the data mover for transfer to a target device for completion.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

BLACK BORDERS

IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

FADED TEXT OR DRAWING

BLURRED OR ILLEGIBLE TEXT OR DRAWING

SKEWED/SLANTED IMAGES

COLOR OR BLACK AND WHITE PHOTOGRAPHS

GRAY SCALE DOCUMENTS

LINES OR MARKS ON ORIGINAL DOCUMENT

REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.